Solution DA mixture

Master DS; academic year 2022-2023

Imputation-Posterior

Fit a two-component mixture of Gaussian distributions with unknown means and variances, the p.d.f is given by

$$f_Y(y) = \pi\phi(y|\mu_1, \sigma_1^2) + (1-\pi)\phi(y|\mu_2, \sigma_2^2),$$

where $\phi(y|\mu, \sigma)$ is the probability density function of a normal distribution with mean μ and variance σ^2, evaluated at y, and $\pi \in (0,1)$. Let $\boldsymbol{\theta} = (\pi, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2)'$.

The complete data $(\boldsymbol{y}, \boldsymbol{z})$ likelihood is given by:

$$p_c(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{\theta}) = \prod_{i=1}^{n} \left[\pi\phi(y_i; \mu_1, \sigma_1^2)\right]^{z_i} \left[(1-\pi)\phi(y_i; \mu_2, \sigma_2^2)\right]^{(1-z_i)}.$$

Consider the following prior

$$\pi(\boldsymbol{\theta}) \propto (\pi(1-\pi))^{-1/2} \sigma_1^{-2} \sigma_2^{-2}.$$

Introduce latent variables

$$z_i|\boldsymbol{\theta} = \begin{cases} 1, & \text{if } y_i \text{ arises from } \phi(\cdot|\mu_1, \sigma_1^2) \text{ with probability } \pi, \\ 0, & \text{if } y_i \text{ arises from } \phi(\cdot|\mu_2, \sigma_2^2) \text{ with probability } 1-\pi. \end{cases}$$

Obtain the complete (augmented) posterior distribution

$$\begin{aligned}
\pi(\boldsymbol{\theta}, \boldsymbol{z}|\boldsymbol{y}) &= \frac{p_c(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{p(y)} \propto p_c(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}) \\
&\propto \pi^{(N_1-1/2)}(1-\pi)^{(N_2-1/2)} \sigma_1^{-N_1-2} \sigma_2^{-N_2-2} \\
&\times \exp\left\{-\frac{1}{2\sigma_1^2} \sum_{i=1}^{n} z_i (y_i - \mu_1)^2\right\} \exp\left\{-\frac{1}{2\sigma_2^2} \sum_{i=1}^{n} (1-z_i)(y_i - \mu_2)^2\right\} \\
&\propto \pi^{(N_1+1/2-1)}(1-\pi)^{(N_2+1/2-1)} (\sigma_1^2)^{-(N_1/2+1)} (\sigma_2^2)^{-(N_2/2+1)} \\
&\times \exp\left\{-\frac{1}{2\sigma_1^2} \sum_{i=1}^{n} z_i (y_i - \mu_1)^2\right\} \exp\left\{-\frac{1}{2\sigma_2^2} \sum_{i=1}^{n} (1-z_i)(y_i - \mu_2)^2\right\}.
\end{aligned}$$

Let $N_1 = \sum_{i=1}^n z_i$, $N_2 = n - N_1$ and remark that the conditional posterior distributions of (μ_1, μ_2) can be written as follows:

$$\pi(\mu_1, \mu_2|\boldsymbol{z}, \boldsymbol{\theta}_{\setminus \mu_1,\mu_2}\boldsymbol{y}) \propto \exp\left\{-\frac{N_1}{2\sigma_1^2}\left(\mu_1 - \frac{1}{N_1}\sum_{i=1}^{n} z_i y_i\right)^2\right\}\exp\left\{-\frac{N_2}{2\sigma_2^2}\left(\mu_2 - \frac{1}{N_2}\sum_{i=1}^{n}(1-z_i)y_i\right)^2\right\}.$$

The full conditional posterior distributions are

$$\begin{aligned}
\pi|\boldsymbol{\theta}_{\setminus}\pi, \boldsymbol{z}, \boldsymbol{y} &\sim Be(1/2 + N_1, 1/2 + N_2),\\
\mu_1|\boldsymbol{\theta}_{\setminus}\mu_1, \boldsymbol{z}, \boldsymbol{y} &\sim \mathcal{N}\left(\frac{1}{N_1}\sum_{i=1}^{n} y_i z_i, \frac{\sigma_1^2}{N_1}\right)\\
\mu_2|\boldsymbol{\theta}_{\setminus}\mu_2, \boldsymbol{z}, \boldsymbol{y} &\sim \mathcal{N}\left(\frac{1}{N_2}\sum_{i=1}^{n} y_i (1-z_i), \frac{\sigma_2^2}{N_2}\right)\\
\sigma_1^2|\boldsymbol{\theta}_{\setminus}\sigma_1^2, \boldsymbol{z}, \boldsymbol{y} &\sim \mathcal{G}^{-1}\left(\frac{N_1}{2}, \frac{1}{2}\sum_{i=1}^{n} z_i (y_i - \mu_1)^2\right)\\
\sigma_2^2|\boldsymbol{\theta}_{\setminus}\sigma_2^2, \boldsymbol{z}, \boldsymbol{y} &\sim \mathcal{G}^{-1}\left(\frac{N_2}{2}, \frac{1}{2}\sum_{i=1}^{n} (1-z_i)(y_i - \mu_2)^2\right)
\end{aligned}$$

Recall the Inverse-Gamma distribution

$$\begin{aligned}
&y \sim \mathcal{G}^{-1}(\lambda, \alpha)\\
&p(y) = \Gamma(\alpha)^{-1}\lambda^{\alpha} y^{-(\alpha+1)} e^{-\lambda/y}.
\end{aligned}$$

Given the observed data y_i, the missing data z_i are Bernoulli with probability π_i,

$$z_i|y_i, \theta \sim Be(\pi_i),$$

where,

$$\begin{aligned}
\pi_i = P\left(z_i = 1|y_i, \boldsymbol{\theta}\right) &= \frac{P\left(z_i = 1, y_i|\boldsymbol{\theta}\right)}{P\left(y_i|\boldsymbol{\theta}\right)}\\
&= \frac{\pi\phi(y_i; \mu_1, \sigma_1^2)}{\pi\phi(y_i; \mu_1, \sigma_1^2) + (1-\pi)\phi(y_i; \mu_2, \sigma_2^2)}.
\end{aligned}$$

Pseudo-algorithm:

- *iteration $t = 0$:* initialization, set $\boldsymbol{\theta}^{(0)}$.

for $t = 1, 2, \ldots$

- *Imputation step*;

  - sample $z_i^{(t)}$ from $z|\boldsymbol{\theta}^{(t-1)}, y_i \sim Bernoulli\left(\pi_i^{(t)}\right),\ 1 \leq i \leq n$, where

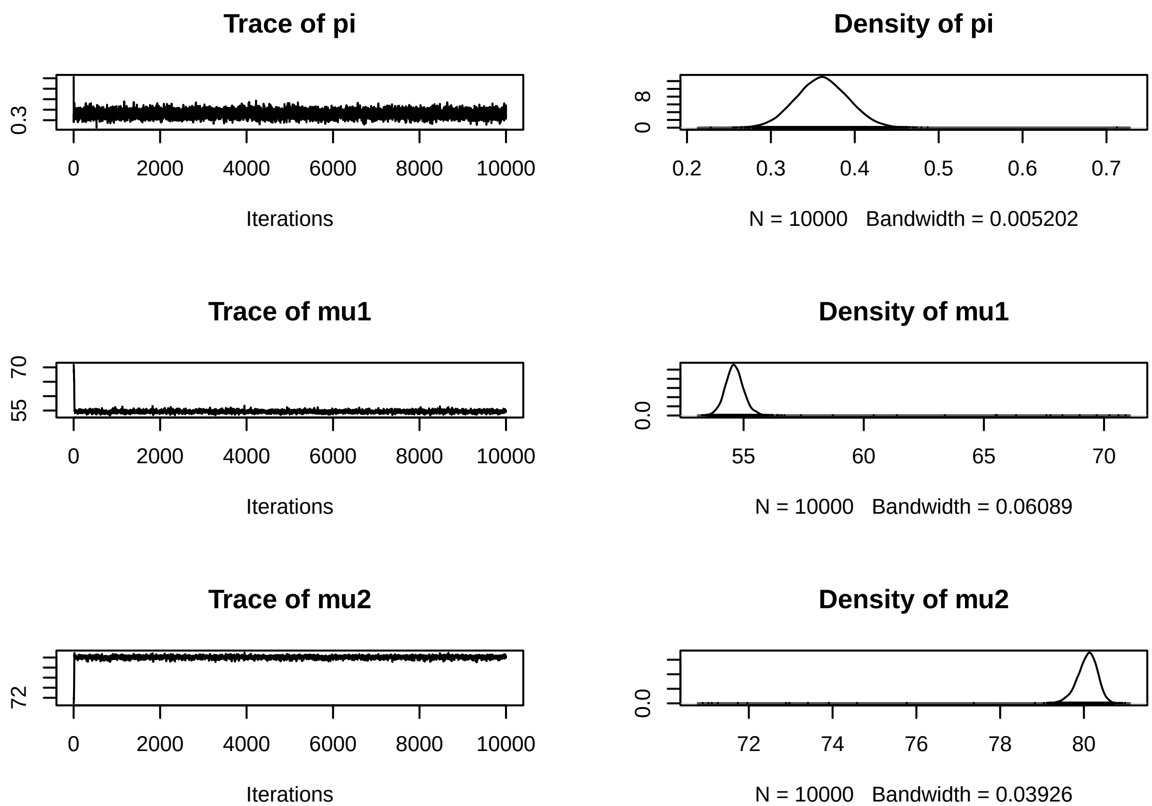
$$\pi_i^{(t)} = \frac{\pi^{(t-1)}\phi\left(y_i; \mu_1^{(t-1)}, \sigma^{2,(t-1)}\right)}{\pi^{(t-1)}\phi\left(y_i; \mu_1^{(t-1)}, \sigma_1^{2,(t-1)}\right) + (1-\pi^{(t-1)})\phi\left(y_i; \mu_2^{(t-1)}, \sigma_2^{2,(t-1)}\right)}.$$

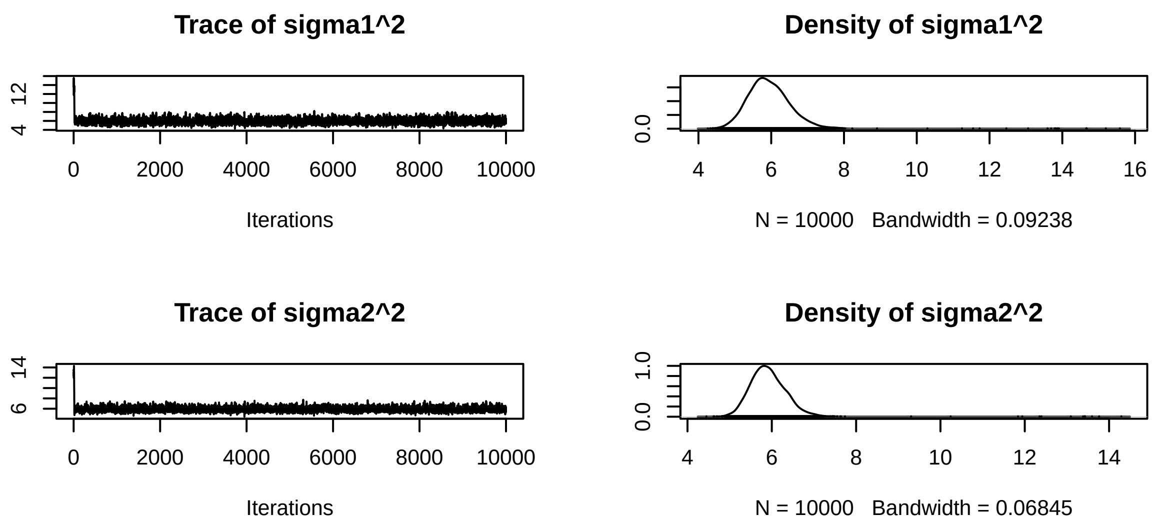
+ Compute $N_1^{(t)} = \sum_{i=1}^{n} z_i^{(t)},\ N_2 = n - N_1^{(t)}$.

- *Posterior step*; sample from the joint posterior distribution $\theta|\boldsymbol{z}, \boldsymbol{y}$
  - sample $\pi^{(t)}$ from $\pi|\boldsymbol{z}^{(t)}, \mu_1^{(t-1)}, \mu_2^{(t-1)}, \sigma_1^{2,(t-1)}, \sigma_2^{2,(t-1)}, \boldsymbol{y} \sim Be(N_1^{(t)} + 1/2, N_2^{(t)} + 1/2)$,
  - sample $\mu_1^{(t)}$ from $\mu_1|\boldsymbol{z}^{(t)}, \pi^{(t)}, \mu_2^{(t-1)}, \sigma_1^{2,(t-1)}, \sigma_2^{2,(t-1)}, \boldsymbol{y} \sim \mathcal{N}\left(\frac{1}{N_1^{(t)}} \sum_{i=1}^n z_i^{(t)} y_i, \frac{\sigma_1^{2,(t-1)}}{N_1^{(t)}}\right)$,
  - sample $\mu_2^{(t)}$ from $\mu_2|\boldsymbol{z}^{(t)}, \pi^{(t)}, \mu_1^{(t)}, \sigma_1^{2,(t-1)}, \sigma_2^{2,(t-1)}, \boldsymbol{y} \sim \mathcal{N}\left(\frac{1}{N_2^{(t)}} \sum_{i=1}^n (1 - z_i^{(t)}) y_i, \frac{\sigma_2^{2,(t-1)}}{N_2^{(t)}}\right)$,
  - sample $\sigma_1^{2,(t)}$ from $\sigma_1^2|\boldsymbol{z}^{(t)}, \pi^{(t)}, \mu_1^{(t)}, \mu_2^{(t)}, \sigma_2^{2,(t-1)}, \boldsymbol{y} \sim \mathcal{G}^{-1}\left(\frac{N_1^{(t)}}{2}, \frac{1}{2}\left(\mu_1^{(t)} - \sum_{i=1}^n z_i^{(t)} y_i\right)^2\right)$,
  - sample $\sigma_2^{2,(t)}$ from $\sigma_2^2|\boldsymbol{z}^{(t)}, \pi^{(t)}, \mu_1^{(t)}, \mu_2^{(t)}, \sigma_1^{2,(t)}, \boldsymbol{y} \sim \mathcal{G}^{-1}\left(\frac{N_2^{(t)}}{2}, \frac{1}{2}\left(\mu_2^{(t)} - \sum_{i=1}^n (1 - z_i^{(t)}) y_i\right)^2\right)$.

```
library(MCMCpack, quietly = TRUE)
library(coda, quietly = TRUE)

data(faithful)
Y<-faithful$waiting
n<-length(Y)
set.seed(123)

M = 10000 #number of MCMC iterations
J = 4 #number of chains
theta = array(dim=c(M,5,J),0)
Z_mat = array(dim=c(M,n,J),0)

for (j in 1:J)
{
  # initialization
  pmix<- rbeta(1,1,1)
  mu0<-c(mean(Y),mean(Y))
  s0<-c(sd(Y),sd(Y))
  theta[1, ,j] = c(pmix, mu0,s0)

  for (i in 2:M)
  {
    # imputation step (generate from conditional posterior dist. of latent variable)
    pc1 = theta[i-1,1, j]*dnorm(Y, theta[i-1,2,j ],theta[i-1,4,j])
    pc2 = (1-theta[i-1, 1,j])*dnorm(Y,theta[i-1,3,j],theta[i-1,5,j])
    Z = rbinom(n,1, pc1 / (pc1 + pc2)  )
    Z_mat[i, ,j] = Z

    theta[i, 1,j] =  rbeta(1,sum(Z)+1/2,sum(1-Z)+1/2)

        if (theta[i, 1, j]>1/2) {
          theta[i, 1,j ] = 1-theta[i, 1,j ]
      Z<- 1-Z
        }
      N1 = sum(Z); N2 = n - N1;

    # posterior - generate posterior from conditional post. dist.
      theta[i, 2,j] = rnorm(1,mean(Y[Z==1]),theta[i-1, 4,j]/N1)
      theta[i, 3,j] = rnorm(1,mean(Y[Z==0]),theta[i-1, 5,j]/N2)
      scale_val = 0.5* sum((Y[Z==1]-theta[i, 2,j])^2)
      theta[i, 4,j] = sqrt(rinvgamma(1,shape = N1/2, scale = scale_val))
```

```
      scale_val = 0.5*sum((Y[Z==0]-theta[i, 3,j])^2)
      theta[i, 5,j] = sqrt(rinvgamma(1,shape = N2/2, scale = scale_val))
  }
}
dimnames(theta) = list(c(), c("pi", "mu1","mu2","sigma1^2","sigma2^2"))
l_mcmc = list(theta[, , 1],theta[, , 2])
plot(as.mcmc(theta[, , 1]))
```






```
apply(theta[, , 1],2 ,mean)
```

```
##        pi        mu1        mu2   sigma1^2   sigma2^2
##  0.3614033 54.6406582 80.0685400  5.9453391  5.9217797
```

```
hpd <- HPDinterval(as.mcmc(theta[, , 1]), 0.95)  ###highest posterior density interval
hpd
```

```
##               lower      upper
## pi        0.3012236  0.4220436
## mu1      53.8346142 55.3832208
## mu2      79.5667836 80.5189308
## sigma1^2  4.8830972  7.0671385
## sigma2^2  5.1671610  6.7488490
## attr(,"Probability")
## [1] 0.95
```

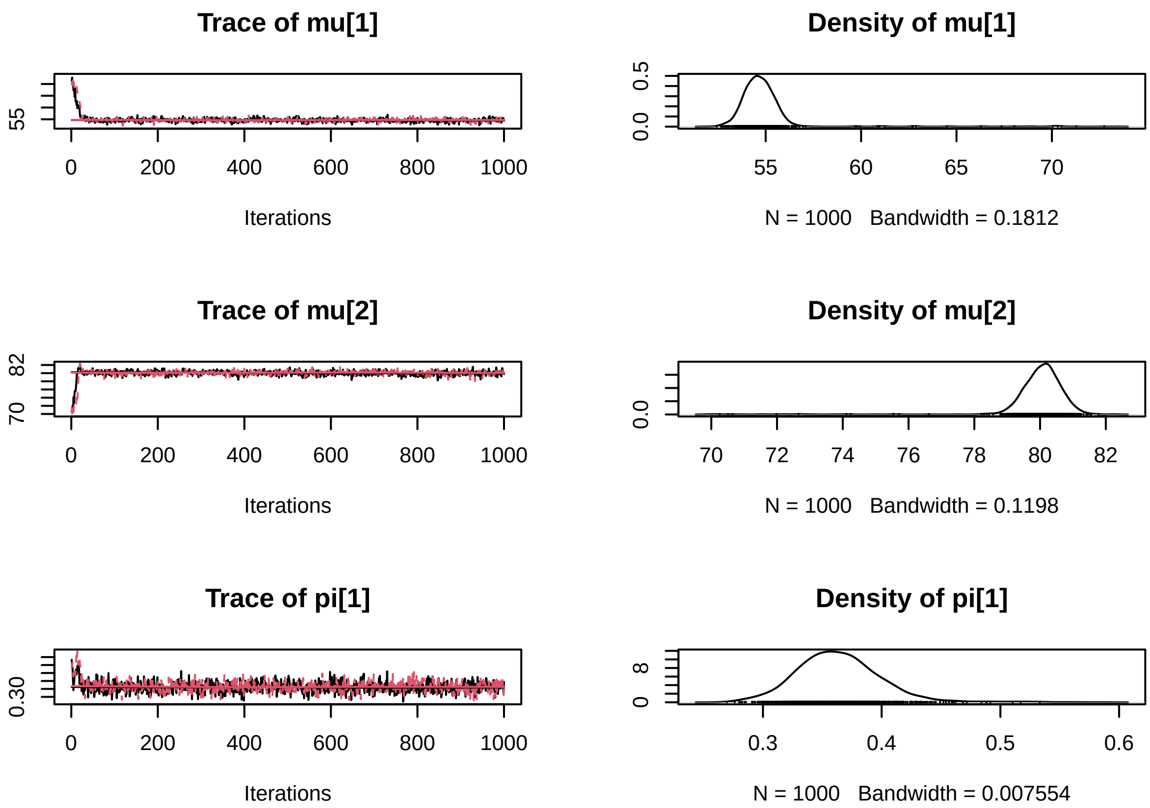
R-Jags

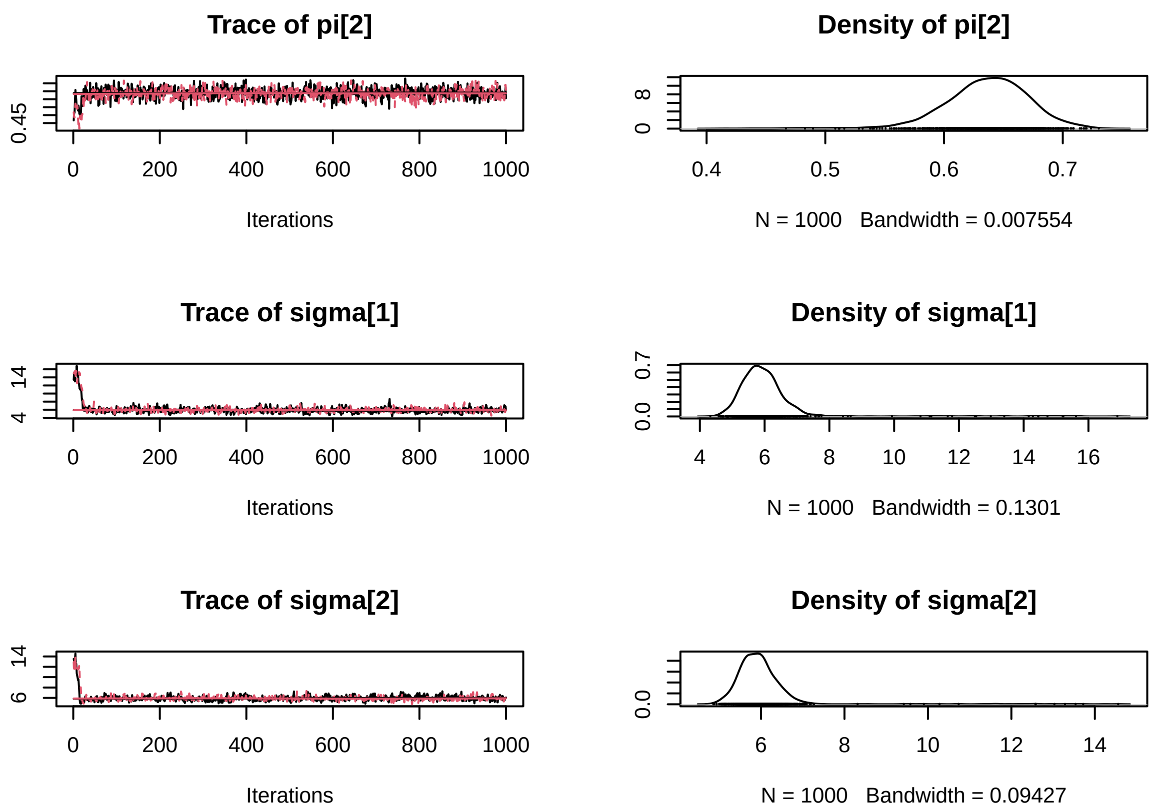
```
library(rjags)
library(R2jags)

jags_model = "
model {
```

```
  for (i in 1:n) {
    y[i] ~ dnorm(mu[zeta[i]], tau[zeta[i]])
    zeta[i] ~ dcat(pi[])
  }

  for (i in 1:H) {
    mu[i] ~ dnorm(0,1e-5)
    tau[i] ~ dgamma(1,1)
    sigma[i] <- 1/sqrt(tau[i])
  }

  pi ~ ddirich(a)
}"


library(datasets)

n=length(faithful$waiting)

dat = list(n=n, H=2, y=faithful$waiting, a=rep(1,2))

jm = jags.model(textConnection(jags_model), data = dat, n.chains = 2)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 272
##    Unobserved stochastic nodes: 277
##    Total graph size: 1105
##
## Initializing model
```

```
r = coda.samples(jm, c('mu','sigma','pi'), 1e3)
plot(r)
```


Trace of mu[1]
55
0
200
400
600
800
1000
Iterations
Density of mu[1]
0.0
0.5
55
60
65
70
N = 1000   Bandwidth = 0.1812
Trace of mu[2]
70
82
0
200
400
600
800
1000
Iterations
Density of mu[2]
0.0
70
72
74
76
78
80
82
N = 1000   Bandwidth = 0.1198
Trace of pi[1]
0.30
0
200
400
600
800
1000
Iterations
Density of pi[1]
0
8
0.3
0.4
0.5
0.6
N = 1000   Bandwidth = 0.007554

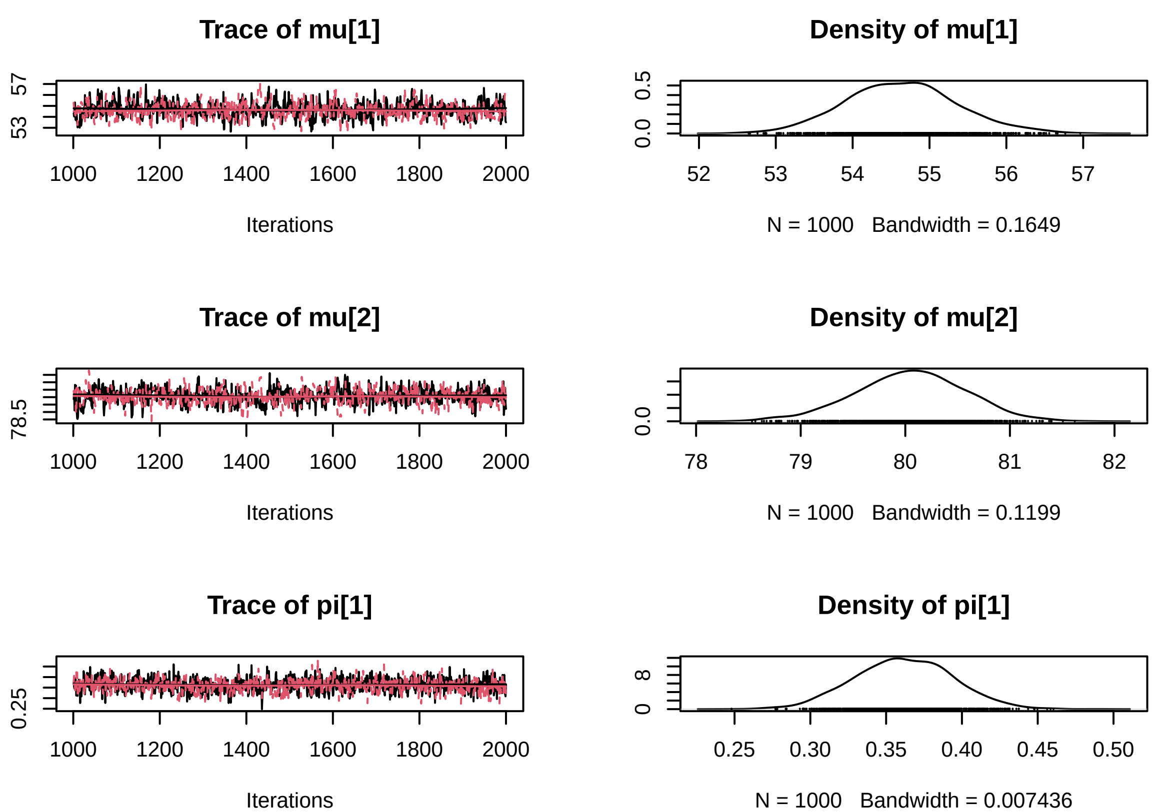


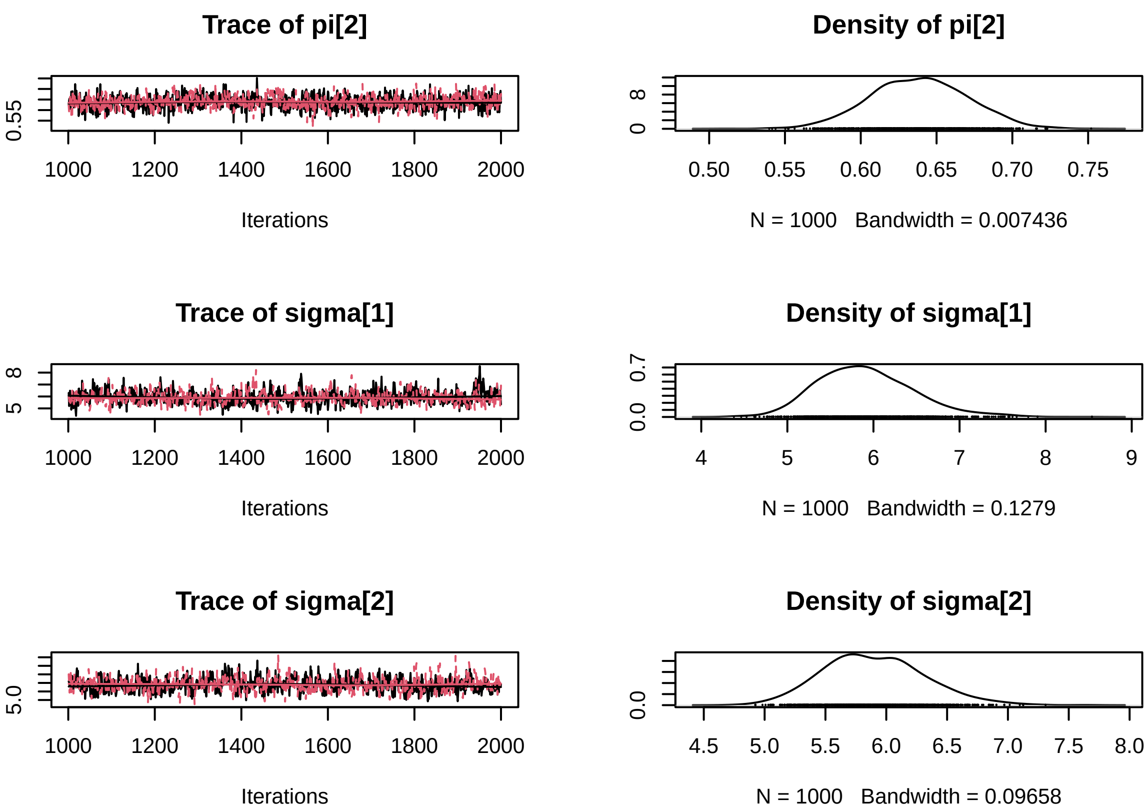

```
jags_model = "
model {
  for (i in 1:n) {
    y[i] ~ dnorm(mu[zeta[i]], tau[zeta[i]])
    zeta[i] ~ dcat(pi[])
  }

  for (i in 1:H) {
    mu0[i] ~ dnorm(0,1e-5)
    tau[i] ~ dgamma(1,1)
    sigma[i] <- 1/sqrt(tau[i])
  }

  mu[1:H] <- sort(mu0)
  pi ~ ddirich(a)
}"


jm = jags.model(textConnection(jags_model), data = dat, n.chains = 2)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 272
##    Unobserved stochastic nodes: 277
```

```
##     Total graph size: 1109
##
## Initializing model
```

```r
r = coda.samples(jm, c('mu','sigma','pi'), 1e3)
plot(r)
```




Trace of pi[2]
0.55
1000 1200 1400 1600 1800 2000
Iterations
Density of pi[2]
0 8
0.50 0.55 0.60 0.65 0.70 0.75
N = 1000 Bandwidth = 0.007436
Trace of sigma[1]
5 8
1000 1200 1400 1600 1800 2000
Iterations
Density of sigma[1]
0.0 0.7
4 5 6 7 8 9
N = 1000 Bandwidth = 0.1279
Trace of sigma[2]
5.0
1000 1200 1400 1600 1800 2000
Iterations
Density of sigma[2]
0.0
4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0
N = 1000 Bandwidth = 0.09658